Self-Sum: Teaching an Agent to Decide Itself *When* and *What* to Summarize

Hongru Wang[α], Rui Wang[γ], Jushi Kai[δ], Boyang Xue[γ], Yongqi Li[τ], Shijue Huang[σ], Xiaoteng Ma[ϵ], Jeff Z. Pan[α], Amos Storkey[α]

[α]University of Edinburgh, [γ]The Chinese University of Hong Kong,
[δ]Shanghai Jiao Tong University, [τ]The Hong Kong Polytechnic University,
[σ]The Hong Kong University of Science and Technology, [ϵ]Tsinghua University
{hrwang, j.z.pan, a.storkey}@ed.ac.uk

Abstract

Long-horizon agents operate over extended sequences of reasoning and actions, but this inevitably accumulates context noise, resulting in excessive computational cost and information overload. Existing approaches commonly rely on fixed, rule-based summarization strategies (e.g., summarizing every few steps), which are inflexible, lack generalization, and often introduce irreversible information loss. We propose *Self-Sum*, a framework that empowers agents to autonomously decide when and what to summarize by modeling summarization as a first-class internal cognitive action, unified with external environmental actions within a multi-turn decision-making process. Specifically, we introduce a two-stage training recipe consisting of (i) a cold-start supervised fine-tuning stage that bootstraps summarization behavior, and (ii) a lightweight, summarization-aware reinforcement learning stage that refines summarization timing and content while discouraging unnecessary summaries. Experiments on multiple long-horizon benchmarks show that *Self-Sum* consistently outperforms no-summarization and rule-based baselines, with particularly strong gains in generalization. Analysis further reveals that *Self-Sum* learns to summarize sparsely at meaningful moments and preserves task-relevant information, highlighting the importance of jointly learning when and what to summarize for robust long-horizon agent behavior[1].

1 Introduction

Large language models (LLMs) have increasingly been deployed as autonomous agents that interact with external environments by interleaving reasoning with actions and observations (Gao et al., 2025; Zhang et al., 2025a; Jin et al., 2025), enabling multi-step interactions, giving rise to long-horizon agents that operate over extended sequences of decisions,

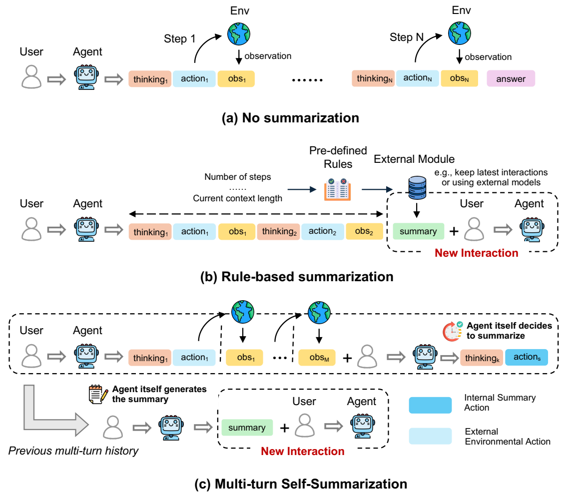

Figure 1: Comparison of context management strategies for long-horizon agents: **(a) No summarization**, which struggles with context grows and eventual context window overflow; **(b) Rule-based summarization** (e.g., ReSum (Wu et al., 2025) or Verl-Agent (Feng et al., 2025)), which follows fixed heuristic rules (e.g., periodic summarization) but risks premature truncation of critical information; and our method **(c) Multi-turn Self-summarization**, where the agent autonomously decides *when* and *what* to summarize across turns, enabling more flexible and coherent information flow.

especially in tool-rich or complex interactive environments (Ye et al., 2025; Wu et al., 2025; Shen et al., 2025). However, as interaction horizons expand, the agent must condition their decisions on an ever-growing history, incurring prohibitive computational cost and introduces unavoidable context noise (Zhou et al., 2025). Consequently, effective context management has emerged as a central challenge (Ye et al., 2025; Lu et al., 2025).

To mitigate context growth, prior work has explored summarization as an effective mechanism for distilling historical interactions into a compact representation (Wu et al., 2025; Lu et al., 2025). Existing approaches largely fall into two categories

[1]The code can be be publicly available via: https://github.com/BayesWatch/SelfSum.

when examined through the lens of *summarization timing* (e.g., when to summarize) and *summarization content* (e.g., what to summarize). As shown in Figure 1, methods without summarization defer compression entirely, implicitly assuming that the full interaction history remains equally relevant throughout decision making (Yao et al., 2022b; Jin et al., 2025). This strategy avoids information loss but provides no control over when summarization should occur, resulting in escalating context noise and computational cost as horizons grow. In contrast, rule-based summarization introduces explicit timing by triggering compression according to predefined heuristics, such as fixed step intervals or context-length thresholds (Wu et al., 2025; Lu et al., 2025). However, these heuristics are agnostic to the agent's internal decision state and task progress, often forcing summarization at suboptimal moments (e.g., too early or too late). More sophisticated variants delegate summarization decisions to an external model that monitors the full trajectory (Wan et al., 2025), but this decouples summarization from the agent's own decision-making process and brings additional inference overhead.

In contrast, humans naturally manage long-horizon problem solving by selectively and proactively summarizing past progress at meaningful milestones, retaining only task-relevant information for future decisions (Miller, 1956; Newell et al., 1972). Inspired by this process, we argue that summarization should be modeled as a *first-class, learnable cognitive action* for the agent, analogous to a human's internal cognitive scratchpad (Miller, 1956). To this end, we propose *Self-Sum*, a framework that enables the agent to decide both the timing and content of summarization itself. Specifically, we reframe long-horizon interactions as a multi-turn conversation between user and agent, and model summary as one special internal cognitive action for agent, unified with external environmental actions within a shared decision space (see Figure 1(c)).

To empower the agent to proactively and selectively summarize current progress, we carefully design a two-stage training recipe: 1) a cold-start supervised fine-tuning stage bootstraps the model to call summary and generate grounding content with less hallucinations in a multi-turn conversation setting; and 2) a summarization-aware reinforcement learning stage further refines the agent's summarization behavior by discouraging unnecessary summary actions while preserving task-relevant information. Through this training procedure, the resulting agent can decide *when* and *what* to summarize itself, along with taking appropriate environmental actions. In summary, the main contributions of this work are as follows.

- We reformulate long-horizon agent interaction by modeling summarization as a *first-class internal cognitive action* unified with external environmental actions in a multi-turn decision process.
- We introduce a simple yet effective two-stage training recipe that enables agents to autonomously decide *when* and *what* to summarize.
- Extensive experiments on two challenging long-horizon benchmarks show that *Self-Sum* consistently outperforms no-summarization and strong rule-based summarization baselines. Notably, *Self-Sum* exhibits substantially stronger generalization across environments and interaction horizons (up to $+2.5$ absolute points), indicating more stable and reliable long-horizon behavior.

2 Related Work

Language Agents. Large language models (LLMs) have rapidly evolved as the core of autonomous agents, with applications spanning tool use (Jin et al., 2025; Qiu et al., 2025; Wang et al., 2025b) and interactive decision-making (Feng et al., 2025; Zhang et al., 2025b). Early work primarily relied on prompting-based methods (e.g., ReAct (Yao et al., 2022b)) or supervised fine-tuning (SFT) to encourage models to autonomously invoke different actions (Qian et al., 2025; Fang et al., 2025b,a). However, recent studies have shown that these approaches often suffer from limited generalization, especially in complex or long-horizon settings (Chu et al., 2025; Li et al., 2026). Consequently, more recent research has shifted toward reinforcement learning (RL) to scale both reasoning and acting of LLM-based agents (Zhang et al., 2025a; Huang et al., 2026; Wang et al., 2025a), with many approaches adopting PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) to enable agents to learn directly from outcome-based rewards (Jin et al., 2025; Wang et al., 2025a). To further improve learning efficiency and stablize optimization, several variants, such as RLOO (Ahmadian et al., 2024), GiGPO (Feng et al., 2025), and RLVMR (Zhang et al., 2025b), further improve learning by providing denser reward signals derived from state information or intermediate reasoning processes. have shown particular effectiveness in long-horizon and interactive environments.

Context Engineering. Effective context management is critical for LLM-based agents to perform long-horizon interactive tasks[2]. Most existing methods assume a single-turn context window and rely on multi-step, interleaved reasoning and acting within that window, which often leads to context overflow and the accumulation of irrelevant or noisy information (Jin et al., 2025; Zhou et al., 2025). To mitigate this issue, recent studies introduce rule-based multi-turn context management (Wang et al., 2023; Wu et al., 2025; Lu et al., 2025), such as summarizing interactions every few steps, which rely on static heuristics and risk premature or irreversible loss of crucial information.

3 Methods

We first formalize our introduced multi-turn long-horizon framework and then present our proposed *Self-Sum* method, which employs a two-stage training paradigm: 1) cold-start supervised fine-tuning (SFT) and 2) summarization-aware reinforcement learning (RL), as shown in Figure 2, to learn *when* to summarize and *what* to summarize automatically.

3.1 Multi-turn Long-horizon Framework

In the multi-turn long-horizon agentic tasks, the agent $\mathcal{M}$ is initized with a question q, an environment $\mathcal{E}$, and an environmental action space $\mathcal{A}_e = \{a_1, ..., a_n\}$ defined by the environment. It needs to interact with the environment or the human user by executing a specific action in $\mathcal{A}$, obtaining the corresponding observation or feedback from $\mathcal{E}$ or human user, and iteratively repeating this processing until the final answer is driven. Without loss of generality, a complete trajectory with T turns between user and agent is defined as follows:

$$\mathcal{H}_T = (U_1, A_1, U_2, A_2, ..., U_T, A_T), \quad (1)$$

where U_i is the user turn which may contain task query or environmental feedback depending on tasks, and A_i denotes the agent turn, which includes the agent's internal reasoning process and the selected action such as `<think>...</think>` `<action>...</action>`. Therefore, each turn represents one atomic action determined by the agent itself, making credit assignment across individual steps significantly easier compared with single turn. We note that this setup also can be easily extended to dialogue and other multi-turn tool-integrated settings, which we leave for future work.

[2]https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents

3.2 SFT Data Acquisition

In order to warm up the model, we design a two-stage data construction pipeline to generate the training dataset: (1) summary and reasoning generation, and (2) multi-turn conversation construction.

Step 1: Summary and Reasoning Generation. Given a successful trajectory $\tau = \{(o_1, a_1), ..., (o_M, a_M)\}$ and o_i, a_i are i-th observation and action, we first process the entire trajectory using an off-the-shelf large language model (e.g., GPT-5-mini). The model is instructed to analyze the trajectory from a global perspective and to determine, at each step, whether a summarization operation should be triggered. For each summary step, the model is then instructed to generates both (i) a concise summary of the progress so far, grounded in observable results rather than inferred or fabricated details; and (ii) an accompanying explanation that justifies why the current step is an appropriate time to summarize. This global trajectory-level processing offers two benefits. First, summaries are generated with access to the full trajectory, thereby avoiding error accumulation that may arise from incremental or stepwise summarization. Second, the generated justifications serve as explicit reasoning signals that supervise the model's summarization-timing decisions. In addition, we instruct the same model to generate action-level reasoning at every step, explaining why the selected action is appropriate given the context (see prompts at Appendix A).

Step 2: Multi-turn Conversation Construction. After obtaining summaries $\{s_1^c, ..., s_M^c\}$, explanations $\{s_1^e, ..., s_M^e\}$, and action rationales $\{r_1, ..., r_M\}$, we reorganize the trajectory into a multi-turn conversational format, where each trajectory is partitioned into several segments according to the number of summarization steps k. Formally, the trajectory is segmented as follows:

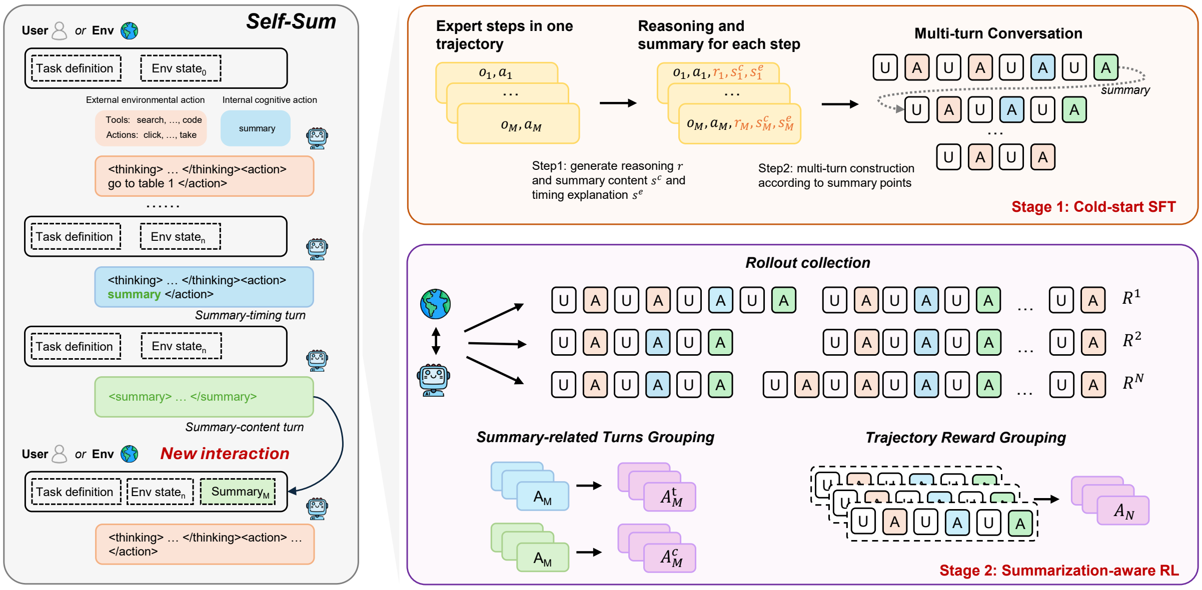

Figure 2: Overview of Self-Sum. The agent interacts with the environment/user via external environmental actions or proactively summarize progress via internal cognitive action, within a unified multi-turn conversation setting. Self-Sum is trained with a two-stage recipe: warm-up SFT and summarization-aware RL.

$$
\begin{aligned}
\mathcal{H}^1 &= \{U^{(0)}, A_1, \dots, U_i, A_i^t, U_{i+1}, A_{i+1}^c\}, \\
\mathcal{H}^2 &= \{U^{(1)}, A_{i+2}, \dots, U_j, A_j^t, U_{j+1}, A_{j+1}^c\}, \\
&\vdots \\
\mathcal{H}^{k+1} &= \{U^{(k)}, A_{t_k+1}, \dots, U_T, A_T^t, U_{T+1}, A_{T+1}^c\}
\end{aligned}
$$

Here, $U^{(0)} = q$ denotes the initial query. For $k \geq 1$, $U^{(k)} = (q, s_k)$ represents the query augmented with the summary s_k generated at k-th summarization step. The U_i denotes observation or feedback by the environment or human during multi-turn interaction. Each turn $(U^{(k)}, ..., U_i, A_i)$ is treated as an independent multi-turn training instance, with loss applied at the final assistant turn. We distinguish three types of assistant actions: 1) *standard turn* (i.e., A_1), corresponding to normal reasoning and chosen action from pre-defined admissible environmental actions; 2) *summarization-timing turn* (i.e., A_i^t), which indicate the decision to initiate summarization with the generated explanations s_k^e; and 3) *summarization-content turn* (i.e., A_{i+1}^c), which generate the summary s_k^c conditioned on the preceding interactions if the summary action is triggered. By explicitly separating summarization timing from summarization content, this formulation enables fine-grained supervision and facilitates more precise learning of summarization behavior within multi-turn interactions. Consequently, we collect around $1,000$ training samples from a small set of successful trajectories (≈ 300) to teach model when is a good timing to summarize and what a good summary looks like. The data statistic analysis can be found in Appendix B.1.

3.3 Summarization-aware RL

To enable the model to proactively summarize its progress during task execution, we augment the original external environmental action space A_e with an additional internal action space $A_s = \{\texttt{summary}\}$, inspired by theory of agents (Wang et al., 2026). Under this formulation, the model follows the same decision-making process to determine when summarization is appropriate itself, rather than a heuristic post-processing step. Once the model selects the `summary` action, it generates the corresponding summary in the subsequent turn. This design allows summarization behavior to be learned end-to-end within the reinforcement learning (RL) framework, as shown in Figure 2.

Episode Relative Advantages We primarily follow GiGPO (Feng et al., 2025) and organize the trajectories and their corresponding returns into an episode-level group to compute relative advantages. Specifically, we collect a group of N trajectories and their corresponding returns for each task:

$$
G^E = \{(\tau_1, \mathcal{R}(\tau_1)), ..., (\tau_N, \mathcal{R}(\tau_N))\}, \quad (2)
$$

where each trajectory $\tau_i = \{\mathcal{H}_i^1, ..., \mathcal{H}_i^T\}$ span the full decision horizon T. We then compute an episode relative advantage for each trajectory to captures whether the agent successfully completes the assignment across the entire episode.

$$\mathcal{A}^E(\tau_i) = \frac{\mathcal{R}(\tau_i) - mean(\{\mathcal{R}(\tau_j)\}_{j=1}^N)}{F_{norm}(\{\mathcal{R}(\tau_j)\}_{j=1}^N)} \quad (3)$$

Summary Relative Advantages While episode-level advantages provide a global learning signal, they are insufficient to guide when and how summaries should be produced. To address this, we introduce fine-grained, summary-specific advantages that operate at the turn level. Specifically, we decompose summary-related rewards into two components: 1) *summarization-timing reward* $\mathcal{R}_s^t$, which evaluates whether summaries are generated at appropriate steps; and 2) *summarization-content reward* $\mathcal{R}_s^c$, which evaluates the quality and usefulness of the generated summaries. If a task is successfully completed and the model generates M summaries, these rewards are evenly distributed across the summary turns:

$$r_{A^t} = \frac{\mathcal{R}_s^t}{M}, \quad r_{A^c} = \frac{\mathcal{R}_s^c}{M} \quad (4)$$

This design encourages the agent to produce summaries more selectively and thoughtfully, promoting careful decisions about both the timing and content of summarization rather than frequent or redundant summaries. Furthermore, we group the summary-related turns within the task and normalize their rewards within that group:

$$\mathcal{A}_{tag}^S = \frac{r_{tag}^S - \mu_{tag}}{\theta_{tag}}, \quad (5)$$

where tag either is summary-timing or summary-content, and μ_{tag} and θ_{tag} are the mean and standard deviation of summary-related rewards for all turns with that specific tag. The final turn-level advantage $\mathcal{A}_t$ is a combination of these two signals:

$$\mathcal{A}_t = \mathcal{A}^E + \mathcal{A}_{tag}^S \quad (6)$$

Finally, we optimize the policy π_θ using a clipped surrogate objective with KL divergence regularization:

$$\mathcal{L}_{\text{final}} = \mathbb{E}_t\big[\min(r_t(\theta)A_t,\ \hat{r}_t A_t)\big] - \lambda_{\text{KL}} D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}), \quad (7)$$

where $\hat{r}_t = \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)$, $r_t(\theta)$ is the importance sampling ratio, and λ_{KL} controls the strength of KL penalty.

4 Experiments

4.1 Set up

Benchmarks. We train the LLM-based agents on two challenging benchmarks: 1) ALFWorld (Shridhar et al., 2021), which is an embodied environment designed to assess the ability of LLM agents to perform multi-step decision-making; and 2) ScienceWorld (Wang et al., 2022), which focuses on text-based scientific experimentation.

Baselines. We compare our approach with a range of competitive baselines: 1) No summarization: GPT-4o, Gemini-2.5-Pro, DeepSeek-V3; 2) Rule-based summarization: **PPO** (Schulman et al., 2017), **RLOO** (Ahmadian et al., 2024), **GRPO** (Shao et al., 2024), **GiGPO** (Feng et al., 2025) based on implementation of verl-agent[3] which always pass latest two observations and actions into next turn; and 3) Multi-turn self-summarization: We re-implement **ReSum** (Wu et al., 2025) under this setting and summarize the progress every 3 steps using the policy model itself; and our proposed ***Self-Sum***.

Implementation Details and Metrics. We largely follow the setting of GiGPO (Feng et al., 2025) to ensure fair comparison and use the pre-collected successful trajectories from Zhang et al. (2025b), we set the maximum prompt length to 2048 tokens to evaluate whether the model can proactively summarize intermediate progress. We also set the summary-related reward as 5 and the agent will get summary rewards only when it successfully complete the task and it calls summary actions. If the prompt exceeds this limit during training, the task is considered a failure, thereby enforcing the policy model to summarize its progress. We run our main experiment with two different seeds and evaluation for three times with different seeds to avoid randomness. We mainly report the success rate for all benchmarks used in our experiments following previous works (Feng et al., 2025).

4.2 Main Results

Table 1 shows the final results. Overall, there are two key important findings.

The effectiveness and robustness of context management depends critically on whether it can adapt both when to summarize and what to

[3] https://github.com/langfengQ/verl-agent

Models	Methods	AlfWorld							SciWorld
		Pick	Look	Clean	Heat	Cool	Pick2	All	Succ.
No Summarization									
GPT-4o	Prompting	75.3	60.8	31.2	56.7	21.6	49.8	48.0	45.4
Gemini-2.5-Pro	Prompting	92.8	63.3	62.1	69.0	26.6	58.7	60.3	-
DeepSeek-R1	ReAct	-	-	-	-	-	-	68.8	22.2
GPT-4o	ReAct	-	-	-	-	-	-	57.3	45.4
Qwen2.5-1.5B-Instruct									
Rule-based	PPO	$64.8_{\pm 3.5}$	$40.5_{\pm 6.9}$	$57.1_{\pm 4.9}$	$60.6_{\pm 6.6}$	$46.4_{\pm 4.0}$	$47.4_{\pm 1.9}$	$54.4_{\pm 3.1}$	-
	RLOO	$88.3_{\pm 3.0}$	$52.8_{\pm 8.6}$	$71.0_{\pm 5.9}$	$62.8_{\pm 8.7}$	$66.4_{\pm 5.5}$	$56.9_{\pm 4.7}$	$69.7_{\pm 2.5}$	-
	GRPO	$85.3_{\pm 1.5}$	$53.7_{\pm 8.0}$	$84.5_{\pm 6.8}$	$78.2_{\pm 7.9}$	$59.7_{\pm 5.0}$	$53.5_{\pm 5.6}$	$72.8_{\pm 3.6}$	21.1
	GiGPO	$94.4_{\pm 5.9}$	$67.5_{\pm 4.6}$	$94.8_{\pm 3.8}$	$\mathbf{94.4}_{\pm \mathbf{7.8}}$	$\mathbf{79.8}_{\pm \mathbf{4.7}}$	$76.4_{\pm 5.4}$	$86.7_{\pm 1.7}$	25.8
Multi-turn	ReSum	$96.2_{\pm 1.4}$	$63.3_{\pm 0.5}$	$\mathbf{100.0}_{\pm \mathbf{0.0}}$	$93.3_{\pm 2.4}$	$63.3_{\pm 4.7}$	$\mathbf{83.3}_{\pm \mathbf{2.1}}$	$86.5_{\pm 0.7}$	$44.0_{\pm 1.3}$
	Self-Sum	$\mathbf{100.0}_{\pm \mathbf{0.0}}$	$\mathbf{86.7}_{\pm \mathbf{4.7}}$	$\mathbf{100.0}_{\pm \mathbf{0.0}}$	$88.3_{\pm 6.2}$	$65.0_{\pm 0.0}$	$\mathbf{83.3}_{\pm \mathbf{2.1}}$	$\mathbf{88.8}_{\pm \mathbf{1.0}}$	$\mathbf{45.1}_{\pm \mathbf{0.4}}$
Qwen2.5-7B-Instruct									
Rule-based	PPO	$92.3_{\pm 4.0}$	$64.0_{\pm 8.4}$	$92.5_{\pm 2.4}$	$89.5_{\pm 7.0}$	$80.3_{\pm 2.0}$	$68.8_{\pm 8.3}$	$80.4_{\pm 2.7}$	46.9
	RLOO	$87.6_{\pm 4.3}$	$78.2_{\pm 8.3}$	$87.3_{\pm 5.8}$	$81.3_{\pm 7.6}$	$71.9_{\pm 5.2}$	$48.9_{\pm 8.4}$	$75.5_{\pm 4.6}$	-
	GRPO	$90.8_{\pm 5.1}$	$66.1_{\pm 6.7}$	$89.3_{\pm 5.4}$	$74.7_{\pm 6.9}$	$72.5_{\pm 5.4}$	$64.7_{\pm 7.3}$	$77.6_{\pm 5.2}$	49.1
	GiGPO	$97.7_{\pm 1.6}$	$82.7_{\pm 7.9}$	$\mathbf{98.8}_{\pm 1.6}$	$83.7_{\pm 7.2}$	$\mathbf{89.3}_{\pm 8.2}$	$79.2_{\pm 6.6}$	$90.8_{\pm 1.3}$	53.4
Multi-turn	ReSum	$99.1_{\pm 1.4}$	$83.3_{\pm 4.7}$	$98.4_{\pm 2.2}$	$90.0_{\pm 0.0}$	$68.3_{\pm 6.2}$	$89.4_{\pm 2.1}$	$89.8_{\pm 1.3}$	$57.8_{\pm 2.3}$
	Self-Sum	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$96.8_{\pm 4.5}$	$\mathbf{95.0}_{\pm 0.0}$	$61.7_{\pm 2.4}$	$\mathbf{95.5}_{\pm 0.0}$	$\mathbf{91.9}_{\pm 0.4}$	$\mathbf{59.1}_{\pm 0.2}$

Table 1: The main results of different summarization-based methods on AlfWorld and SciWorld. Some results are directly copied from Feng et al. (2025), we **bold** the best score among all models.

summarize across environments. Across benchmarks and models, *Self-Sum* consistently outperforms no-summarization and strong rule-based summarization baselines, as well as other multi-turn summarization strategies. This result underscores the importance of learning adaptive summarization policies, rather than relying on fixed heuristics, to handle diverse task structures and interaction horizons. We observe remaining weaknesses on specific sub-tasks (e.g., *Cool* and *Clean*), which are likely attributable to SFT data imbalance and repetitive action patterns; we analyze these cases further in § 5.4.

Different environments impose different requirements on summarization timing and content. The relative effectiveness of summarization strategies varies substantially across environments. On AlfWorld, where task structures are relatively simple and long-term dependencies are limited, rule-based methods (e.g., GiGPO) can match or even outperform multi-turn ReSum, suggesting that fixed heuristics are sometimes sufficient. In contrast, SciWorld demands longer-term state tracking and tighter coupling between early observations and later decisions. In this regime, rigid summarization rules frequently fail to preserve critical information. As a result, ReSum yields substantial gains over other baselines, and our proposed *Self-Sum* further improves performance by enabling the agent to decide autonomously when to summarize and what information to retain.

Reward	AlfWorld		SciWorld	
	Succ.	#Sum.	Succ.	#Sum.
Self-Sum	88.8	2.91	45.1	1.49
w/o SFT	48.4	0.38	7.8	0.05
w/o RL	5.2	4.77	3.9	1.53
w/o summary reward	88.2	2.81	42.5	1.47

Table 2: The ablation study results on Qwen2.5-1.5B-Instruct model. We additionally report the average number of `summary` action used during the inference.

5 Analysis

In this section, we provide detailed analysis and case study to showcase the advantages of our proposed method and current bottlenecks to guide future work.

5.1 Ablation Study

Effects of SFT and RL. We conduct a controlled ablation study by removing key components of *Self-Sum* to analyze the contribution of each part. Table 2 reports the results. Removing SFT significantly degrades performance and almost eliminates the use of the summarization action. This indicates that SFT is essential for teaching the model when to summarize and how to summarize. In contrast, removing RL leads to poor performance with more summary action used, which suggests RL primarily refines the content of summarization. In addition, the fine-grained summarization reward design further regulate the agent's summarization behavior, leading to improved performance. To-

Reward	AlfWorld		SciWorld	
	Succ.	#Sum.	Succ.	#Sum.
1	84.3	2.85	37.5	0.59
3	86.5	2.67	41.2	0.67
5	88.8	2.91	45.1	1.49

Table 3: The effects of different value of summary rewards across two benchmarks. We report both sucess rate and average number of summary calls.

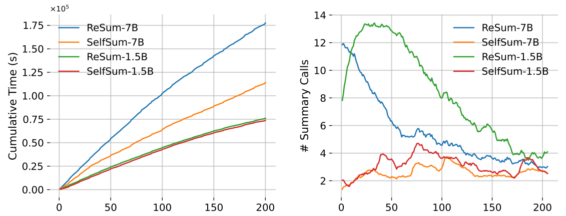

Figure 3: The training dynamics analysis on AlfWorld. **Left)**: Cumulative wall-clock time as a function of training steps. Lower curves indicate higher time efficiency under the same step budget; **Right)**: Average number of summary actions invoked per step over training, reflecting the agent's summarization behavior.

gether, these results demonstrate that effective long-horizon agents requires learning both what to summarize and when to summarize, and that SFT and RL play complementary roles in enabling these two capabilities.

Effects of Reward. We additionally set the summary timing and content reward as different numbers (e.g., 1,3,5) to investigate the effects. Table 3 shows the results. It can found that increasing the number of reward tends to encourage the model to use more summary actions, leading to better performance, while the reward exceeds 5 does not bring significant differences in our experiments.

5.2 Behavior Analysis

Efficiency Analysis. We also draw the training dynamics of timing and summary actions as shown in Figure 3. On the left side, it is evident that our proposed *Self-Sum* consistently cost less cumulative training time than ReSum under the same step budget, indicating improved training efficiency across model scales, especially for larger model. On the right side, ReSum exhibits heavy summary usage at the beginning of training, which gradually decreases over time, whereas *Self-Sum* maintains fewer and more stable summary invocations throughout training. These trends suggest that our proposed *Self-Sum* learns to call summarization more selectively rather than frequently, leading to both reduced computational overhead and more efficient training.

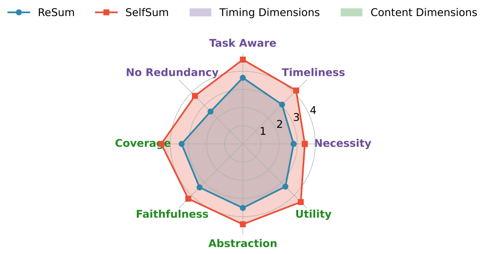

Figure 4: The comparison of summarization behavior between *Self-Sum* and ReSum, in terms of both summarization timing and content from 4 different dimensions respectively.

Summary Analysis. We randomly sample 50 task-matched trajectories from the outputs of *Self-Sum* and ReSum, and employ an LLM-as-a-judge (Zheng et al., 2023) to evaluate the quality of the generated summaries along both timing and content dimensions. Specifically, we instruct GPT-5-mini to score summarization behavior in terms of timing (e.g, Necessity, Timeliness, Task Awareness, and Avoidance of Redundancy) and content (e.g., Coverage of Critical Information, Faithfulness, Level of Abstraction, and Future Utility). The full evaluation prompt is provided in the Appendix A.2. As shown in Figure 4, *Self-Sum* consistently outperforms the strong baseline across all evaluated dimensions, demonstrating clear advantages in both when to summarize and what to summarize.

5.3 More Results

Out-of-domain Evaluation. Table 4 reports the generalization performance on unseen tasks in AlfWorld and SciWorld under increasing levels of distribution shift. As expected, all methods experience substantial performance degradation as task variants and categories become fully unseen (e.g., L1 $\rightarrow$ L2), reflecting the inherent difficulty of transferring long-horizon policies to novel task structures. Further analysis on failure cases reveal the model tends to generate relatively more hallucinations for these unseen tasks. Overall, *Self-Sum* achieves the strongest generalization performance across settings, with particularly clear advantages in the more challenging SciWorld environment.

Results on Webshop. We additionally implement *Self-Sum* on WebShop using the Qwen2.5-1.5B-Instruct backbone, with results reported in Table 5. It is found that *Self-Sum* consistently

Methods	AlfWorld			SciWorld		
	L0	L1	L2	L0	L1	L2
DS-V3	60.2	65.9	53.9	27.3	35.2	26.5
DS-R1	68.8	70.2	67.3	22.2	31.4	29.1
Qwen2.5-1.5B-Instruct						
GRPO	76.6	71.1	29.7	21.1	13.7	10.9
GiGPO	86.7	83.2	48.0	25.8	15.2	4.7
ReSum	86.5$_{\pm 0.7}$	83.3$_{\pm 0.4}$	-	44.0$_{\pm 1.3}$	35.9$_{\pm 1.7}$	35.2$_{\pm 1.7}$
SelfSum	88.8$_{\pm 1.0}$	84.4$_{\pm 0.2}$	48.4$_{\pm 0.2}$	45.1$_{\pm 0.4}$	36.2$_{\pm 0.9}$	36.5$_{\pm 1.3}$
Qwen2.5-7B-Instruct						
GRPO	79.3	77.3	52.3	49.1	30.1	26.6
GiGPO	89.5	**90.2**	67.2	53.4	35.2	25.8
ReSum	89.8$_{\pm 1.3}$	85.2$_{\pm 1.3}$	-	57.8$_{\pm 2.3}$	46.4$_{\pm 0.1}$	41.8$_{\pm 0.1}$
Self-Sum	**91.9**$_{\pm 0.4}$	87.5$_{\pm 2.3}$	**68.2**$_{\pm 3.0}$	**59.1**$_{\pm 0.2}$	**49.0**$_{\pm 0.6}$	**44.3**$_{\pm 0.2}$

Table 4: The generalization results on unseen tasks in ALFWorld and SciWorld: 1) L0: seen task variants and categories as same as main experiment; 2) L1: unseen task variants but seen task categories; and 3) L2: unseen task variants and categories. DS means DeepSeek.

Methods	WebShop	
	Score.	Succ.
GPT-4o	31.8	23.7
Gemini-2.5-Pro	42.5	35.9
AgentLM-7B (Zeng et al., 2024)	-	36.5
AgentEvol (Xi et al., 2024)	-	76.5
GiGPO-1.5B	83.5$_{\pm 1.8}$	67.4$_{\pm 4.5}$
GiGPO-7B	86.2$_{\pm 2.6}$	75.2$_{\pm 3.8}$
Self-Sum	**89.2**$_{\pm 0.7}$	**77.1**$_{\pm 0.5}$

Table 5: The results of *Self-Sum* on WebShop (Yao et al., 2022a) with Qwen-2.5-1.5B-Instruct.

outperforms prior prompting-based and rule-based summarization methods by a clear margin on both score and success rate. Notably, despite using a smaller 1.5B model, *Self-Sum* surpasses several strong baselines built on 7B backbones (e.g., AgentEvol (Xi et al., 2024) and GiGPO-7B), achieving the highest average score (89.2) and success rate (77.1) with low variance across runs. These results indicate that Self-Sum generalizes effectively to complex, real-world environments such as web navigation, highlighting the robustness of proactive, model-driven summarization for long-horizon decision making.

5.4 Others

Error Analysis. Figure 6 presents a breakdown of failure cases across AlfWorld and SciWorld. Only a small fraction of failures are attributable to summary-related issues, such as omitting critical task information or producing inaccurate state descriptions (e.g., object status), reflecting the effectiveness of our training recipe. The majority of errors stem from repetitive or invalid actions that cannot be executed by the environment. This indicates that agents often fall into action loops or repeatedly execute redundant steps without making substantive progress. Together, effective long-horizon control calls for more fine-grained credit assignment between internal summarization decisions and external environmental actions, in addition to optimizing the timing and content of summaries, which we leave to future work.

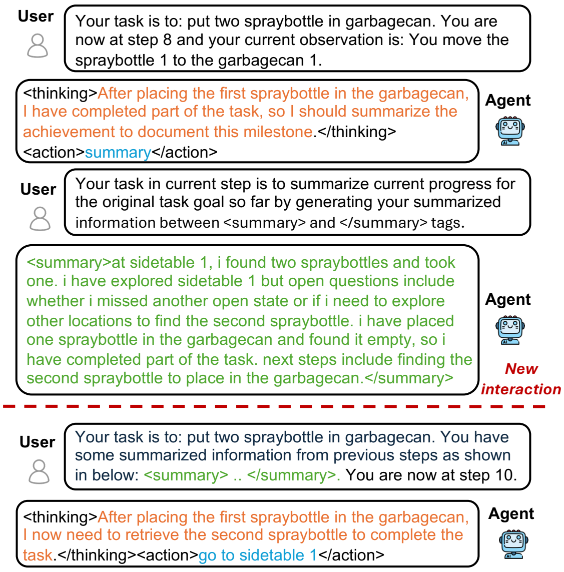

Figure 5: A trajectory segment of *Self-Sum* in AlfWorld. We use different color to indicate thinking, action, generated summaries.

Case Study. We randomly sample a successful trajectory of *Self-Sum* on AlfWorld and provide a partial segment in Figure 5. It can be observed that *Self-Sum* proactively tracks task progress and identifies appropriate moments to summarize, such as at a meaningful milestone (Step 8). At this point, the agent generates a well-grounded summary that accurately captures completed sub-goals and remaining uncertainties, while also outlining plausible next steps. Crucially, this summary is later effectively reused to guide subsequent decision-making. When the interaction is reinitialized (Step 10), the agent correctly conditions on the summarized state without relying on the full raw trajectory, and selects the appropriate action to continue the task. Overall, this case study demonstrates that *Self-Sum* supports coherent long-horizon behavior by coupling timely summarization with informed future actions through effective reuse of summarized context.

6 Conclusion

In this paper, we propose *Self-Sum*, a framework that reformulates long-horizon agent interaction as a multi-turn decision process and models summarization as a first-class internal cognitive action unified with environmental actions. This design enables agents to autonomously decide *when* and *what* to summarize with a carefully designed two-stage training recipe, allowing context compression to be learned rather than imposed by fixed heuristics. Experimental results on ALFWorld, SciWorld and WebShop demonstrate that *Self-Sum* consistently outperforms no-summarization and rule-based summarization baselines, with particularly strong gains in generalization to unseen tasks.

Acknowledgments

We would like to thank the anonymous reviewers and the area chairs for their constructive and insightful feedback, which has significantly improved this paper.

Limitations

Following previous methods (Feng et al., 2025), we adopt the same backbone models from the Qwen2.5 series to ensure a fair and controlled comparison. Our implementation choices are guided by a focus on isolating the effects of the proposed summarization framework, rather than scaling model size or introducing additional components. While larger backbone models may further improve absolute performance, our approach is inherently model-agnostic and can be readily extended as computational resources permit. In addition, we deliberately avoid outsourcing summarization to separate or more powerful models, as doing so would delegate summarization decisions to external modules and obscure the goal of learning summarization as an integral part of the agent's own decision-making process.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand. Association for Computational Linguistics.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*.

Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. 2025a. Webevolver: Enhancing web agent self-improvement with co-evolving world model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8970–8986.

Tianqing Fang, Zhisong Zhang, Xiaoyang Wang, Rui Wang, Can Qin, Yuxuan Wan, Jun-Yu Ma, Ce Zhang, Jiaqi Chen, Xiyun Li, Hongming Zhang, Haitao Mi, and Dong Yu. 2025b. Cognitive kernel-pro: A framework for deep research agents and agent foundation models training. *Preprint*, arXiv:2508.00414.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for llm agent training. *Preprint*, arXiv:2505.10978.

Huanang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Qihan Ren, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang. 2025. A survey of self-evolving agents: On path to artificial super intelligence. *Preprint*, arXiv:2507.21046.

Shijue Huang, Hongru WANG, Wanjun Zhong, Zhaochen Su, Jiazhan Feng, Bowen Cao, and Yi R. Fung. 2026. Adactrl: Towards adaptive and controllable reasoning via difficulty-aware budgeting. *Transactions on Machine Learning Research*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

Mukai Li, Qingcheng Zeng, Tianqing Fang, Zhenwen Liang, Linfeng Song, Qi Liu, Haitao Mi, and Dong Yu. 2026. Verified critical step optimization for LLM agents. *CoRR*, abs/2602.03412.

Miao Lu, Weiwei Sun, Weihua Du, Zhan Ling, Xuesong Yao, Kang Liu, and Jiecao Chen. 2025. Scaling llm multi-turn rl with end-to-end summarization-based context management. *arXiv preprint arXiv:2510.06727*.

George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.

Allen Newell, Herbert Alexander Simon, et al. 1972. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ.

Cheng Qian, Emre Can Acikgoz, Hongru Wang, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. SMART: Self-aware agent for tool overuse mitigation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4604–4621, Vienna, Austria. Association for Computational Linguistics.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, Xing Zhou, Dongrui Liu, Ling Yang, Yue Wu, Kaixuan Huang, Shilong Liu, Hongru Wang, and Mengdi Wang. 2025. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *Preprint*, arXiv:2505.20286.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Junhong Shen, Hao Bai, Lunjun Zhang, Yifei Zhou, Amrith Setlur, Shengbang Tong, Diego Caples, Nan Jiang, Tong Zhang, Ameet Talwalkar, and Aviral Kumar. 2025. Thinking vs. doing: Agents that reason by scaling test-time interaction. *Preprint*, arXiv:2506.07976.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Guangya Wan, Mingyang Ling, Xiaoqi Ren, Rujun Han, Sheng Li, and Zizhao Zhang. 2025. Compass: Enhancing agent long-horizon reasoning with evolving context. *Preprint*, arXiv:2510.08790.

Hongru Wang, Cheng Qian, Manling Li, Jiahao Qiu, Boyang Xue, Mengdi Wang, Heng Ji, Amos Storkey, and Kam-Fai Wong. 2026. Position: Agent should invoke external tools only when epistemically necessary. *Preprint*, arXiv:2506.00886.

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. 2025a. Acting less is reasoning more! teaching model to act efficiently. *Preprint*, arXiv:2504.14870.

Hongru Wang, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam-Fai Wong. 2023. Cue-CoT: Chain-of-thought prompting for responding to in-depth dialogue questions with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12047–12064, Singapore. Association for Computational Linguistics.

Rui Wang, Ce Zhang, Jun-Yu Ma, Jianshu Zhang, Hongru Wang, Yi Chen, Boyang Xue, Tianqing Fang, Zhisong Zhang, Hongming Zhang, Haitao Mi, Dong Yu, and Kam-Fai Wong. 2025b. Explore to evolve: Scaling evolved aggregation logic via proactive online exploration for deep research agents. *Preprint*, arXiv:2510.14438.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. 2025. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2024. Agentgym: Evolving large language model-based agents across diverse environments. *Preprint*, arXiv:2406.04151.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022a. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022b. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. 2025. Agentfold: Long-horizon web agents with proactive context management. *Preprint*, arXiv:2510.24699.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2024. Agenttuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3053–3077.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita-Velez, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. 2025a. The landscape of agentic reinforcement learning for llms: A survey. *Preprint*, arXiv:2509.02547.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. 2025b. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. *Preprint*, arXiv:2507.22844.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*.

Data	AlfWorld	SciWorld
# trajs	300	300
# samples	978	1199
# summaries	678	899
# assistant turns	4,051	6,721

Table 6: The data statistics analysis for both AlfWorld and SciWorld.

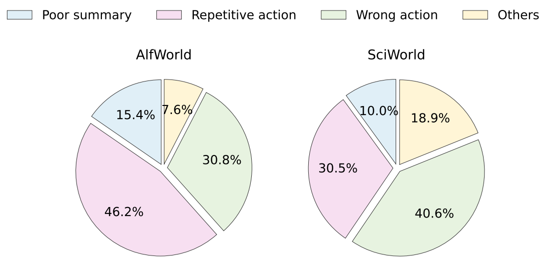

Figure 6: Error analysis results.

A Prompt Templates

A.1 Data Collection

Table 11 shows the specific prompt to generate required summaries and corresponding explanations. Specifically, we explicitly require the model make such decisions carefully and ground the content of summary into observable results. Therefore, the collected SFT data can effectively showcases what a good summary looks like and when is a good timing to summarize.

Table 11 shows the specific prompt to instruct the model generate reasoning processing for each step/action. Therefore, the model can unfold its own thinking processing to support the determined action in each step.

A.2 Other Prompts

Table 8 shows the system prompt to instruct the model to make decisions during training, and Table 7 shows the prompt to generate the summaries.

Table 10 shows the prompt used to evaluate the quality of generated summaries.

B Supplementary Details

B.1 Data Statistics

Table 6 shows the data statistic analysis for our collected SFT data. We only leverage randomly sampled 300 successful trajectories and successfully collect around 1,000 training samples for both datasets.

Please ignore previous instruction, and your task in current step is to summarize current progress for the original task goal so far by generating your summarized information between <summary> and </summary> tags.

Your original task goal is to: {task_description} You are now at step {current_step} and your current observation is: {current_observation}
Your output should start with <summary> and end with </summary>.

Table 7: The system prompt used to instruct the model to generate summary during training.

You are an expert agent operating in the ALFRED Embodied Environment. You need to carefully review all information received, reason about current states, make required actions in order to complete pre-defined task at the environment. You may be given the summarized information from your previous steps for reference.

Now it's your turn to take an action.
You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Table 8: The system prompt used to instruct the model to make decisions during training.

You are an expert agent operating in the ALFRED Embodied Environment. I will provide you with a successful trajectory. You need to supplement the reasoning process.
Reason using ONLY ONE tag pair and express your reasoning in one concise, brief sentence:

You need to output a list in JSON format, with the same length as the trajectory. Each element should contain two key-value pairs, for example:
"""json ["reason": "<thinking>The book may be in the cabinet, shelf, so in the next steps I need to search these locations.</thinking>", "action": "go to shelf 1",
"reason": "<thinking>Currently, my sub-goal is to obtain item A. I have already spotted A, and in order to accomplish this objective, I need to pick it up.</thinking>", "action": "pick up A"]
"""
The "action" field must match the action in the trajectory, and the "reason" field should be a reasonable reasoning process inferred from the context of previous actions and the next few actions.
now the trajectory is as follows: {traj}

Table 9: The system prompt to generate reasoning for each step in the successful trajectory.

You are an expert evaluator of long-horizon language agents. Your task is to assess the timing and content quality of summaries produced during an agent's interaction with an environment or user.

You will be given a full trajectory of the agent's interaction with the environment or user, including the summary steps. You need to carefully read the reasoning process of agent, especially when the agent decides to summarize and generates the summary itself. Your evaluation should focus on when the summary is generated and what information it retains, rather than on task success alone.

Evaluation Criteria

1. Summary Timing (When to Summarize): Evaluate whether summarization is triggered at an appropriate moment.
(a) Necessity: Was summarization necessary at this point, given the interaction history? Could the agent reasonably continue without summarizing?
(b) Timeliness: Is the summary generated too early (before sufficient information is accumulated), too late (after relevant context has already become noisy), or at an appropriate milestone?
(c) Task Awareness: Does the timing reflect awareness of task structure (e.g., subgoal completion, resolution of uncertainty, or transition to a new phase)?
(d) Avoidance of Redundancy: Does the agent avoid summarizing repeatedly without meaningful progress since the last summary?

2. Summary Content Quality (What to Summarize): Evaluate the quality and usefulness of the summarized information.
(a) Coverage of Critical Information: Does the summary retain information that is essential for future decisions (e.g., completed subgoals, important observations, object states)?
(b) Faithfulness to Observations: Is the summary grounded in the interaction history, without introducing unsupported or speculative details?
(c) Abstraction Level: Does the summary appropriately abstract away low-level details while preserving task-relevant structure? Is it overly concrete or overly vague?
(d) Future Utility: Would this summary meaningfully help an agent make correct decisions in subsequent steps?

You output should be in JSON format. The score should be a number between 1 and 5. The justification should be a brief explanation of your score.
"""json {{ "summary_timing": {{ "necessity": <score>, "timeliness": <score>, "task_awareness": <score>, "avoidance_of_redundancy": <score> }}, "summary_content_quality": {{ "coverage_of_critical_information": <score>, "faithfulness_to_observations": <score>, "abstraction_level": <score>, "future_utility": <score> }}, "overall_score": <score>, "justification": "<justification>" }} """

The trajectory is as follows:
{trajectory}

Table 10: The prompt used to evaluate the quality of summaries.

You are analyzing a trajectory of actions taken to complete an ALFRED task. Your job is to decide when it would be beneficial to create a summary of past progress.

Task: {task}

Full Trajectory: {trajectory}

For EACH step in this trajectory, decide:
1. Should we create a summary AFTER this step? (yes/no) considering the observation in current step and previous steps.
2. If yes, provide a concise summary that:
- Summarizes what has been accomplished and what has been observed so far (locations visited, objects found, actions taken, etc.)
- Is grounded ONLY in the observations and actions already taken (NO hallucination or assumptions)
- Mentions key items/locations discovered that may be relevant for the task
- Provides brief guidance for what should be done next based on the task goal and what's been learned
- Uses past tense for completed actions and present/future tense for next steps

Guidelines for when to summarize:
- After completing a significant sub-task or milestone (e.g., found a key object, finished exploring an area)
- When the history is getting long (e.g., after 6-10 steps without a summary)
- When there's a natural transition point (e.g., finished searching, now need to manipulate objects)
- At least once every 5 steps if no other criteria are met
- Considering the observation in current step and all information in in previous steps. Do not consider the action in current step.

Output format (JSON array, must have exactly {num_steps} elements):
{ "step": 1,
"observation": "observation text",
"action": "action taken",
"should_summarize": false,
"explanation": "",
"summary": "" },
{ "step": 2,
"observation": "observation text",
"action": "action taken",
"should_summarize": true,
"explanation": "explanation for why I choose to summarize here",
"summary": "Progress: I have explored countertop 1..."
},
.....

CRITICAL RULES:
1. The "observation" and "action" fields must EXACTLY match the trajectory
2. Summaries must ONLY mention what has been directly observed or done - no speculation
3. Include specific object and receptacle IDs that were found (e.g., "apple 1 on countertop 2")
4. Use summaries strategically - too many summaries are counterproductive
5. Do not summarize consecutively.

Table 11: The system prompt used to instruct the model to generate summary explanation and specific summaries for summary steps. We provide specific guidelines and critical rules for the model to decide good timing to summarize and generate high-quality summaries for later decisions.